Claims

1. A method for processing a multiplicity of data update requests made by a customer, said method comprising the steps of:

grouping all of said data update requests into a plurality of blocks for execution by a data processor, the data update requests within each of said blocks and from one of said blocks to a next one of said blocks being arranged in an order that said data update requests need to be executed to yield a proper data result, each of said blocks having approximately a same capacity for said data update requests, said capacity corresponding to a number of said data update requests which said data processor can efficiently process in order before processing said data update requests in the next one of said blocks; and

then said data processor processing said data update requests within said one block in said order, and then said data processor processing said data update requests within said next block in said order.

2. A method as set forth in claim 1 wherein said order is an order in which said data update requests were made.

3. A method as set forth in claim 1 wherein:

said capacity corresponds to a number of said data update requests which said data processing unit can optimally process in order in said one block before processing said data update requests in said next one of said blocks.

4. A method as set forth in claim 1 wherein said data update requests within each of said blocks are arranged into said order by order information stored within or associated with said blocks.

5. A system for processing a multiplicity of data update requests made by a customer, said method comprising the steps of:

means for grouping all of said data update requests into a plurality of blocks for execution by a data processor, the data update requests within each of said blocks and from one of said blocks to a next one of said blocks being arranged in an order that said data update requests need to be executed to yield a proper data result, each of said blocks having approximately a same capacity for said data update requests, said capacity corresponding to a number of said data update requests which said data processor can efficiently process in order before processing said data update requests in the next one of said blocks;

said data processor including means for then processing said data update requests within said one block in said order;

said data processor including means for then processing said data update requests within said next block in said order.

6. A system as set forth in claim 5 wherein said order is an order in which said data update requests were made.

7. A system as set forth in claim 5 wherein:

said capacity corresponds to a number of said data update requests which said data processing unit can optimally process in order in said one block before processing said data update requests in said next one of said blocks.

8. A computer program product for processing a multiplicity of data update requests made by a customer, said computer program product comprising:

a computer readable medium;

first program instructions to group all of said data update requests into a plurality of blocks for execution by a data processor, the data update requests within each of said blocks and from one of said blocks to a next one of said blocks being arranged in an order that said data update requests need to be executed to yield a proper data result, each of said blocks having approximately a same capacity for said data update requests, said capacity corresponding to a number of said data update requests which said data processor can efficiently process in order before processing said data update requests in the next one of said blocks; and

second program instructions within said data processor to then process said data update requests within said one block in said order, and third program instructions within said data processor to then process said data update requests within said next block in said order; and wherein

said first, second and third program instructions are recorded on said medium.

9. A computer program product as set forth in claim 8 wherein said order is an order in which said data update requests were made.

10. A computer program product as set forth in claim 8 wherein:

said capacity corresponds to a number of said data update requests which said data processing unit can optimally process in order in said one block before processing said data update requests in said next one of said blocks.

11. A method for processing a multiplicity of first data update requests made by a first customer and a multiplicity of second data update requests made by a second customer, said method comprising the steps of:

grouping all of said first data update requests into a plurality of first blocks for execution by a first data processor unit, the first data update requests within each of said first blocks and from one of said first blocks to a next one of said first blocks being arranged in a first order that said first data update requests need to be executed to yield a proper data result, each of said first blocks having approximately a same first capacity for said first data update requests, said first capacity corresponding to a number of said first data update requests which said first data processing unit can efficiently process in order before processing said first data update requests in the next one of said first blocks, and then said first data processing unit processing said first data update requests within said one first block in said first order, and then said first data processing unit processing said first data update requests within said next first block in said first order; and

grouping all said second data update requests into a plurality of second blocks for execution by a second data processor unit, the second data update requests within each of said second blocks and from one of said second blocks to a next one of said second blocks being arranged in a first order that said second data update requests need to be executed to yield a proper data result, each of said second blocks having approximately a same second capacity for said second data update requests, said second capacity corresponding to a number of said second data update requests which said second data processing unit can efficiently process in order before processing said second data update requests in the next one of said second blocks, and then said second data processing unit processing said second data update requests within said one second block in said second order, and then said second data processing unit processing said second data update requests within said next second block in said second order.

12. A method as set forth in claim 11 wherein said first order is an order in which said first data update requests were made, and said second order is an order in which said second data update requests were made.

13. A method as set forth in claim 11 wherein:

said first capacity corresponds to a number of said first data update requests which said first data processing unit can optimally process in order in said one first block before processing said first data update requests in the next one of said first blocks; and

said second capacity corresponds to a number of said second data update requests which said second data processing unit can optimally process in order in said one second block before processing said second data update requests in the next one of said second blocks.

14. A method as set forth in claim 11 wherein said first data processing unit processes said first data update requests in parallel with said second data processing unit processing said second data update requests.

15. A method for updating a database based on data update requests, said method comprising the steps of:

determining an order in which said data update requests were made;

grouping said data update requests into blocks, wherein each of said blocks includes a plurality of data update requests and information indicating said order to process each of said data update requests;

generating and storing information indicating a sequence in which to execute said blocks based on said order; and

processing each of said data update requests within all of said blocks based on said order by a same processor.

16. A method as set forth in claim 15 wherein each of said blocks has a same predetermined capacity for data update requests.